



**Critical**  
manufacturing  
an ASM PT company

# Containers

## 11.3

April 2026

### DOCUMENT ACCESS

Public

### DISCLAIMER

The contents of this document are under copyright of Critical Manufacturing S.A. it is released on condition that it shall not be copied in whole, in part or otherwise reproduced (whether by photographic, or any other method) and the contents therefore shall not be divulged to any person other than that of the addressee (save to other authorized offices of his organization having need to know such contents, for the purpose for which disclosure is made) without prior written consent of submitting company.

# Containers

*Estimated time to read: 9 minutes*

## Overview

A **Container** is a standardized physical carrier used to hold, protect, organize, and move **Materials**, parts, or other containers throughout manufacturing and logistics processes.

A Container acts as the interface between production, storage, and transportation. It ensures that items can be handled safely, efficiently, and repeatedly across the shop floor.

Manufacturers design containers to match:

- The size and weight of the contents
- The sensitivity of the Materials
- The handling methods used on the shop floor, such as manual handling, forklifts, conveyors, AGVs, or robots

Industrial containers can be:

- Returnable or disposable
- Standard or custom-designed
- Made from plastic, metal, wood, or composite materials

Beyond simple storage, containers support lean manufacturing principles by enabling:

- Visual management
- Work-in-process control
- Ergonomic handling
- Quality protection
- Traceability

A container is therefore not just a box. It is a critical logistics and production asset that connects material flow, safety, efficiency, and quality across the industrial shop floor.

## Examples

<b>FOUP for Semiconductor Wafers</b>	<b>SMT Board Magazine</b>



**Pallet**

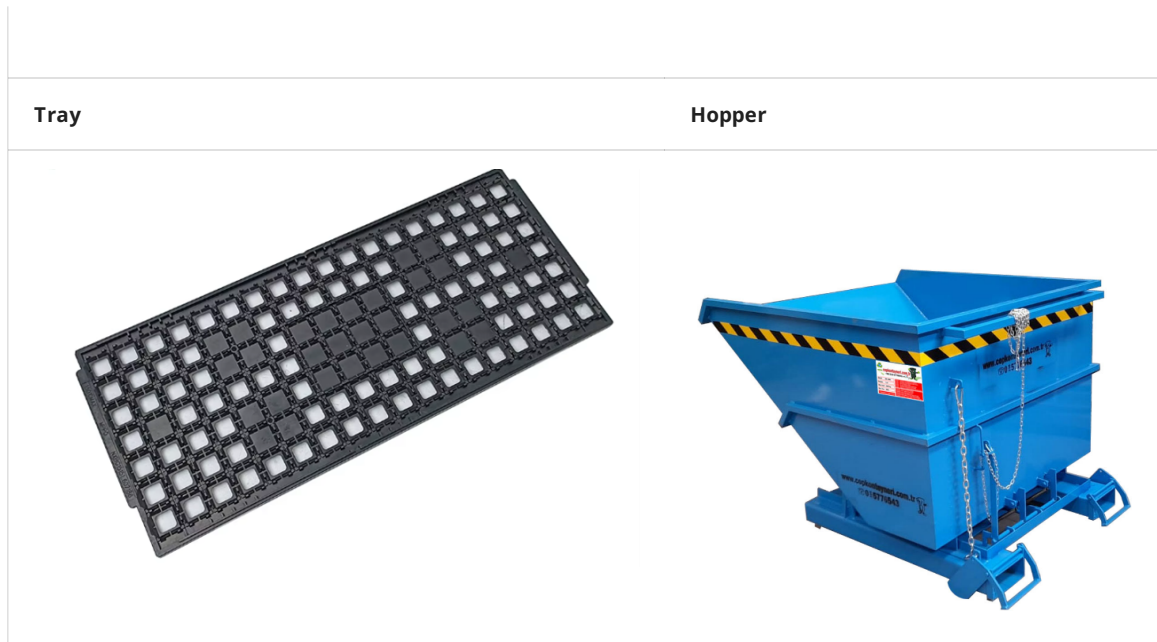
**Trolley of Boxes**



**SMT Feeders Cart**

**ESD Box**





## Containers in MES

A **Container** is an object of the **Container** entity that can hold **Materials**, **Resources**, or other **Containers**.

A **Material Container** has a capacity defined by positions available for a given **Material** unit. A **Container** can hold multiple **Materials**, but each position can contain only one **Material**.

Each position:

- Is identified by a unique number
- Has a defined capacity based on specific units

The same concept applies to:

- **Containers** that hold **Resources**
- **Containers** that hold other **Containers** (SubContainers)

flowchart TD

```
SC[Sub Container]
C[Container]
```

```
F[Facility]
MC[Material Container]
CR[Container Resource]
```

```
M[Material]
R[Resource]
```

```
SC --> C
```

```
C --> F
C --> MC
C --> CR
```

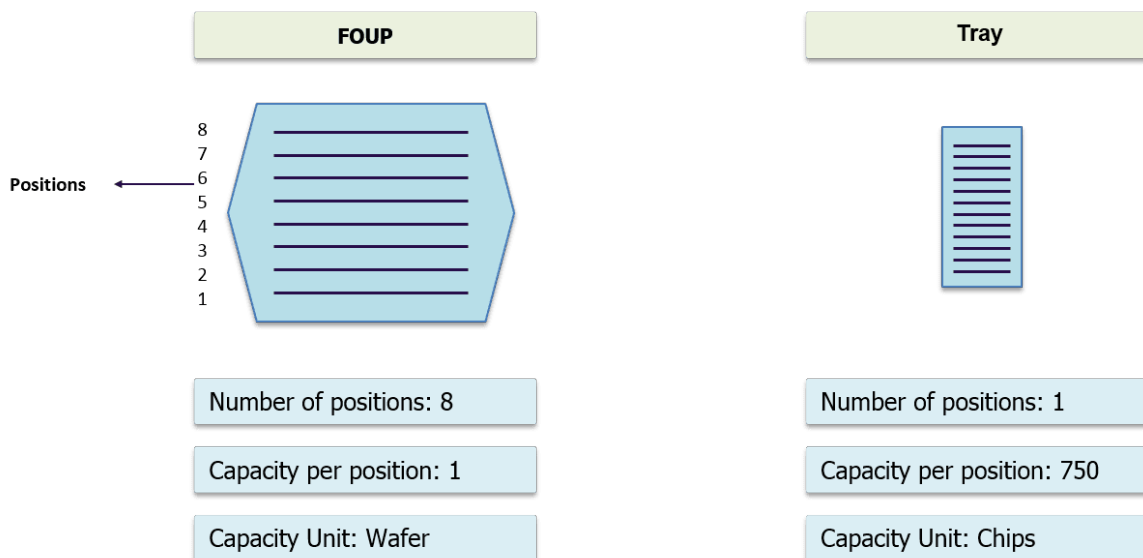
```
MC --> M
CR --> R
```

```
classDef mermaid_title color:#000, fill:#fafafa, stroke:#fafafa, stroke-width:0x, font-size:100%, font-weight:200;
classDef mermaid_start color:#000, fill:#fafafa, stroke:#fafafa, color:#fafafa, stroke-
```

```
width:0x, font-size:100%, visibility: hidden;
classDef mermaid_businessdata color:#000, fill:#65CDE8, stroke:#65CDE8, stroke-width:0px, font-size:100%;
classDef mermaid_nonbusinessdata color:#000, fill:#B7DEE8, stroke:#B7DEE8, stroke-width:0px, font-size:100%;
classDef mermaid_entity color:#000, fill:#FB9F53, stroke:#FB9F53, stroke-width:0px, font-size:100%;
classDef mermaid_entitylinked color:#000, fill:#FCD5B5, stroke:#FCD5B5, stroke-width:0px, font-size:100%;
classDef mermaid_context color:#000, fill:#B9CDE5, stroke:#B9CDE5, stroke-width:0px, font-size:100%;
classDef mermaid_optional color:#000, fill:#B7DEE8, stroke:#65CDE8, stroke-width:1px, font-size:100%, stroke-dasharray: 5 5;
class C,SC mermaid_entity
class F,M,P,R mermaid_businessdata
class MC,CR mermaid_nonbusinessdata
```

A **Container** can be modeled in different ways:

- **Magazine** — one position with multiple units per position (for example, a tray holding several units of the same Material)
- **FOUP (Front-Opening Universal Pod)** — multiple positions with one unit per position (for example, a wafer carrier)
- **Hybrid models** — combinations of both approaches



A **Container** can also be associated with a **Resource** when the Resource supports the position unit types defined in that Container type.

## State Model

Containers have a system-defined state model:

- **Available** — the Container can be used (empty or full)
- **Unavailable** — the Container cannot be used
- **In Transit** — the Container is shipping Materials to another Facility

```
graph TD
  Start -->|Create Container| A1
  A1[Available] -->|Make Container Unavailable| A2[Unavailable]
  A2 -->|Make Container Available| A1
  A1 -->|Ship Material| L1[In Transit]
  L1 -->|Unship Material<br>Receive Material| A1
```

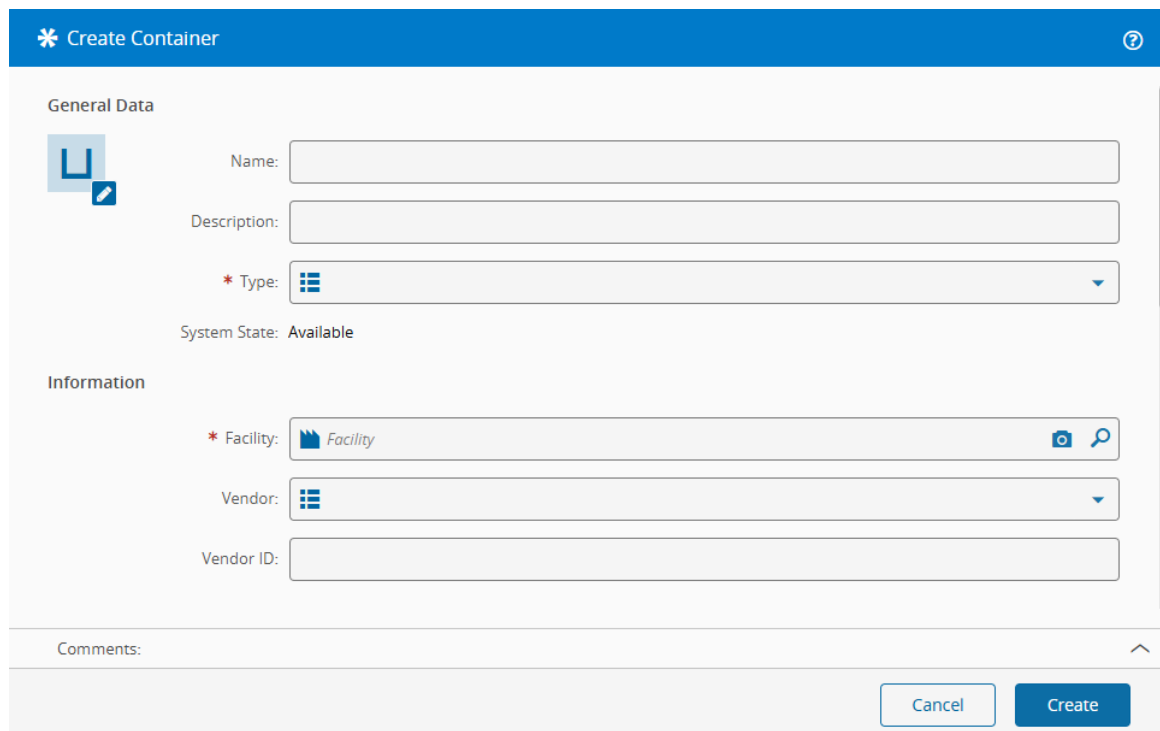
```
classDef mermaid_start color:#000, fill:#fafafa, stroke:#fafafa, color:#fafafa, stroke-width:0x,
font-size:100%, visibility: hidden;
classDef mermaid_entitylinked color:#000, fill:#FCD5B5, stroke:#FCD5B5, stroke-width:0px, font-
size:100%;
classDef mermaid_state color:#000, fill:#d7e4bd, stroke:#000, stroke-width:1px, font-size:100%,
font-weight:300;
class Start mermaid_start
class A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12 mermaid_state
class L1,L2,L3,L4,L5,L6 mermaid_entitylinked
```

You can also create custom **State Models** for Containers.

## Container Creation

The Container creation wizard includes several required fields:

- **Name** — identifies the Container
- **Type** — used in configuration tables for container restrictions
- **Facility** — defines where the Container will be used



Additional configuration fields include:

- **Position Unit Type** — defines whether the Container holds Materials, Containers, or Resources
- **Total Positions** — defines the number of usable positions
- **Capacity Validation Mode** — available only for Material containers

If the **Capacity Validation Mode** is set to **Unit**, you can define the maximum Material quantity per position using **Capacity per Position**.

### Info

For more information, refer to the User Guide: [Create Container](#)

**\* Create Container** ?

Positions

Position Unit Type:  Material  
 Container  
 Resource

Auto Generated:

Unlimited:

\* Total Positions:

\* Capacity Validation Mode:  x ▾

\* Capacity Unit:  ▾

\* Capacity per Position:

\* Orientation:  x ▾

Position Sorting:  Ascending  
 Descending

Comments: ^

## Container Operations

### Manage Positions

Use **Manage Positions** to add or remove objects from Container positions.

The operation works the same way for all **Container Position Unit Types**:

- Material
- Container
- Resource

For **Material** or **Container**:

**Container Positions**

Container of Material (Available)

Container Positions (Used 3/100)

- Cookie01
- Cookie02
- Cookie03** ×
- Empty
- Empty
- Empty
- Empty

Position Details

Material:

Product:

Product Description: Belgas Cookies

Step:  Mixing

Primary Quantity: 20 Kg

Secondary Quantity:

Comments:

For **Container of Containers**, define the relationship between the container and sub-container in the **Container Type Relation** Generic Table.

Container Type Relation (1)

<input type="checkbox"/>	MAIN CONTAINER TYPE	SUB CONTAINER TYPE
<input type="checkbox"/>	Trolley	Box

**Container Positions**

Container of Container (Available)

Container Positions (Used 1/100)

- Container of Material** ×
- Empty
- Empty
- Empty
- Empty
- Empty
- Empty

Position Details

Container:

Description:

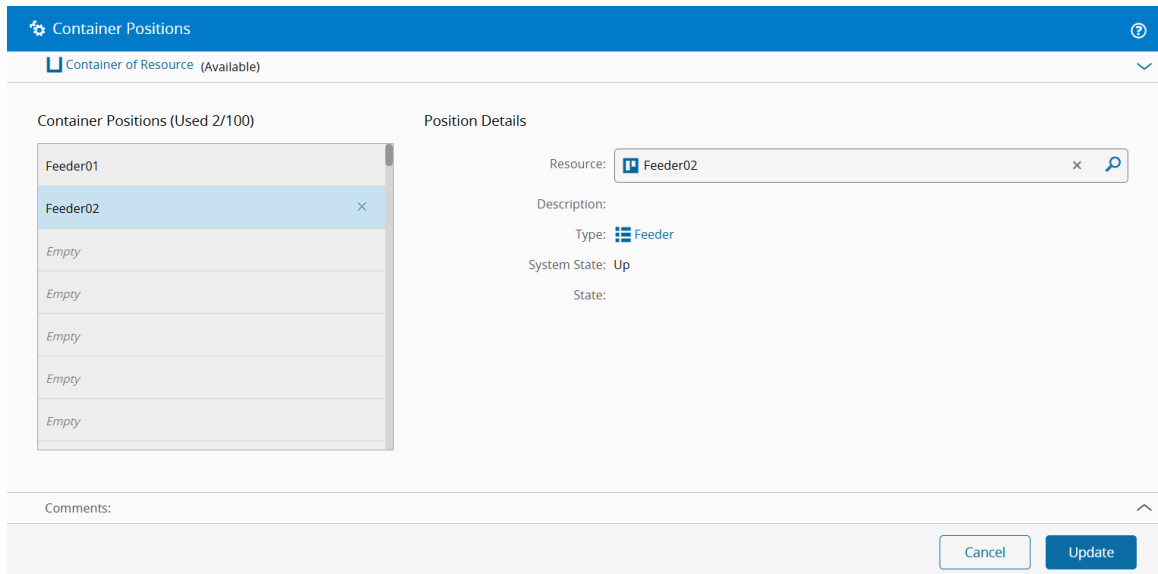
Type:  Box

Comments:

For **Container of Resources**, define the relationship in the **Container Resource Relation** Generic Table.

Container Resource Relation (1)

<input type="checkbox"/>	CONTAINER TYPE	RESOURCE TYPE
<input type="checkbox"/>	Table	Feeder



## Empty

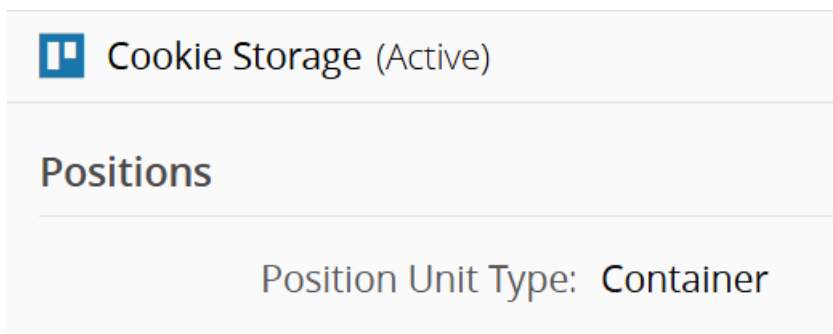
The **Empty** operation removes all objects from the Container.

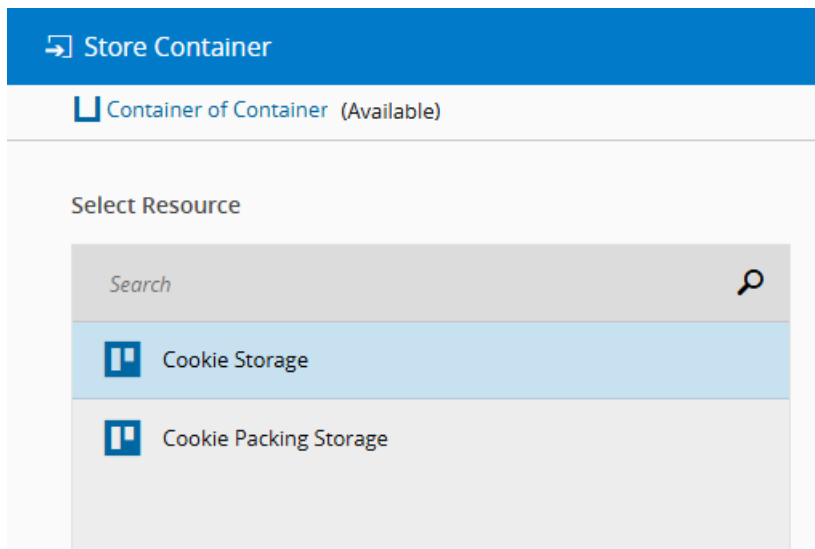
## Store

The **Store** operation places the Container in a **Storage Resource** located in any **Area** that belongs to the Container **Facility**.

Precondition:

The Storage Resource must have **Position Unit Type = Container**.





## Retrieve

The **Retrieve** operation removes the Container from storage.

## Dock

Use **Dock** to dock a Container to a **Resource**.

Docking can occur on Resources with the following processing types:

- **Load Port**
- **Process**

### Load Port Docking

Requirements:

- The Resource must have a **Load Port Type** selected
- The **Position Unit Type** must be **Container**

**Edit Resource** ?

GENERAL DATA ADDITIONAL INFORMATION

**Load Port Information**

\* Load Port Type:  x ▾

**Positions**

\* Position Unit Type:  x ▾

Container Type:  ▾

Track Positions:

Auto Generate Positions:

Total Positions:

**Other Information**

Enable Request MAO on State Change:

Resource Group:

Comments: ^

**Example:**

In SMT lines, loader and unloader machines handle magazines of electronic boards.



### Process Docking

Requirements:

- The Resource must have **Locations enabled**
- A valid **Location** must be selected

■ Dock Container ?

■ Cart1 (Available) ∨

**Resource**

\* Resource: ■ CM-Resource A × 🔍

\* SMT: ■ Loc1 × 🔍

Description:

System State: Up

Area: ■ CM-Area

Priority: 5

Used Positions: 0

Total Positions: 10,000

Comments: ^

Cancel
Dock

When docking a **Container of Resources** (for example feeder carts), the system automatically considers the Resources as consumable feeds in the docking Resource.

**Example:**


In SMT lines, component placement machines can handle multiple feeder carts.





The docking Resource must have:

- **Has Resource Locations** enabled
- **Location Type** defined
- **Dimensions** configured

 Edit Resource

GENERAL DATA   ADDITIONAL INFORMATION   LOCATIONS

**Locations**

Has Resource Locations:

Location Type:

\* Dimension X Size:

\* Dimension Y Size:

Locations must include:

- Location
- Position Count
- Position Consumable Feed Type
- Enabled status

✎ Edit Resource
?

GENERAL DATA
ADDITIONAL INFORMATION
LOCATIONS

Locations + 🗑️

Loc1 <small>Description</small>	🗑️
Loc2 <small>Description</small>	
Loc3 <small>Description</small>	
Loc4 <small>Description</small>	

### Locations Details

Name: Loc1

Description:

Barcode ID:

\* Location X:

\* Location Y:

\* Position Count:

\* Position Consumable Feed Type: ☰ Feeder x ▼

Enabled:

Comments: ^

Cancel
Save

## Undock

The **Undock** operation removes the Container from the Resource.

🗑️ Undock Container ?

🗑️ Cart1 (Available) ▼

**Resource**

Name: 🗑️ CM-Resource A

SMT: 🗑️ Loc1

Description:

System State: Up

Area: 🗑️ CM-Area

Priority: 5

Used Positions: 0

Total Positions: 10,000

Comments: ^

Cancel
Undock

## Add

Use the **Add** operation to assign the current Container to a **Parent Container**.

Define the relationship in the **Container Type Relation** Generic Table.

Container Type Relation (1)	
MAIN CONTAINER TYPE	SUB CONTAINER TYPE
<input type="checkbox"/> Trolley	Box

🗑️ Add Container to Container ?

🗑️ Container of Material (Available) ▼

**Container Details**

\* Container: 🗑️ Container of Container x 🔍

Description:

Type: 🗑️ Trolley

Position: Auto Generated ▼

Used Positions: 0/100

Vendor Id:

Comments: ^

Cancel
Add

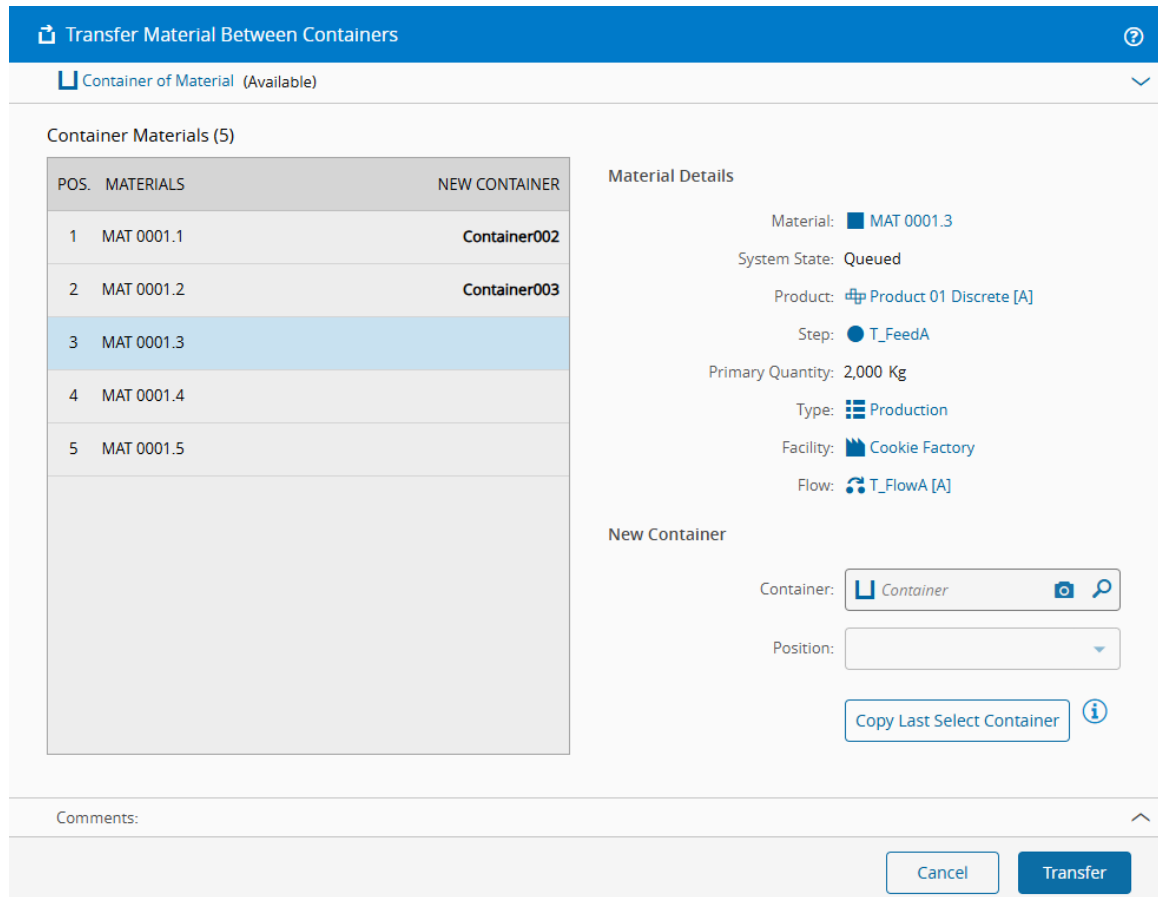
## Transfer

The **Transfer** operation transfers the contents of a Container to another Container.

The operation name varies depending on the **Container Position Unit Type**:

- Transfer Materials
- Transfer Resources
- Transfer Containers

You can transfer each position to a different Container.



**Transfer Material Between Containers**



Container of Material (Available)

POS.	MATERIALS	NEW CONTAINER
1	MAT 0001.1	Container002
2	MAT 0001.2	Container003
3	MAT 0001.3	
4	MAT 0001.4	
5	MAT 0001.5	


**Material Details**

Material: MAT 0001.3  
 System State: Queued  
 Product: Product 01 Discrete [A]  
 Step: T\_FeedA  
 Primary Quantity: 2,000 Kg  
 Type: Production  
 Facility: Cookie Factory  
 Flow: T\_FlowA [A]

**New Container**

Container:   

Position:



Comments:

## Make Unavailable

Sets the **System State** of the Container to **Unavailable**. In this state, the Container is not visible in selection lists.

## Make Available

Changes the Container **System State** back to **Available**.

## Container View

The **Container View** displays the Container layout and contents.

You can:

- Visualize positions and assigned objects
- View details of the selected position

- Customize visible columns in **View Options**

The screenshot shows the 'Container of Material' view. The main area contains a table with the following data:

Parent Material	Material	Product	Step	Quantity
1	Cookie01	Biscuits (A)	Mixing	20 Kg
2	Cookie02	Biscuits (A)	Mixing	20 Kg
3	Cookie03	Biscuits (A)	Mixing	20 Kg
4	Cookie04	Biscuits (A)	Mixing	20 Kg
5	Cookie05	Biscuits (A)	Mixing	20 Kg
6	Cookie06	Biscuits (A)	Mixing	20 Kg
7	Cookie07	Biscuits (A)	Mixing	20 Kg

The right sidebar, titled 'VIEW OPTIONS', has the following toggles:

- Parent Material:
- Material:
- Material Type:
- Product:
- Product Description:
- Product Group:
- Step:
- Flow:
- Facility:
- Quantity:

## Container Maintenance View

The **Container Maintenance View** displays scheduled and ad-hoc **Maintenance Action Orders (MAO)** related to the Container.

For more information, refer to [Maintenance Management](#).

The screenshot shows the 'Container of Material' view in maintenance mode. The main area displays a calendar for January 2026. The date 14th is highlighted in yellow. The right sidebar, titled 'Activities', shows the following sections:

- USAGE BASED (0)
  - No usage based activities
- AD-HOC (0)
  - No ad-hoc activities


## Material Container Type Context

The **Material Container Type Context** Smart Table defines restrictions for container usage.

You can restrict container types based on:

- **Area**
- **Step**
- **Product**
- **Product Group**
- **Material Type**
- **Material Form**
- **Material State**
- **Capacity Class**

You can also define the **maximum number of usable positions** for a Container type.

 **Note**


To select a Step in this Smart Table, enable the Use Container Restrictions property.

### Example

Example configuration using **Step**, **Material Type**, and **Container Type**.

Material Container Type Context (1)									
AREA	STEP	PRODUCT	PRODUCT G	MATERIAL TYPE	MATERIAL FORM	MATERIAL STATE	MATERIAL CAPAC	CONTAINER TYPE	CONTAINER MAX
	CM-StepA			Production				Box	

In this configuration, when you attempt to add a **Material** of type **Production** at the selected Step, only Containers of type **Box** appear in the selection list.

 Add Material to Container ?

SMT Test1 (Queued) / CM-Product A [A] / CM-StepA / 500 EA

**Container Details**

\* Container:

Position:


- Container of Material
- Container of Material1
- Container of Material2
- Container of Material3
- Container of Material4

5 results Advanced

Comments:


Cancel Add

**Container**



Name: **Container of Material**

Description:

Type:  **Box**

Universal State: **Active**

System State: **Available**



# Legal Information

## **Disclaimer**

The information contained in this document represents the current view of Critical Manufacturing on the issues discussed as of the date of publication. Because Critical Manufacturing must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Critical Manufacturing, and Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only.

Critical Manufacturing makes no warranties, express, implied or statutory, as to the information herein contained.

## **Confidentiality Notice**

All materials and information included herein are being provided by Critical Manufacturing to its Customer solely for Customer internal use for its business purposes. Critical Manufacturing retains all rights, titles, interests in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Critical Manufacturing and the Customer must not disclose or transfer by any means any of these materials or information, whether total or partial, to any third party without the prior explicit consent by Critical Manufacturing.

## **Copyright Information**

All title and copyrights in and to the Software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

## **Trademark Information**

Critical Manufacturing is a registered trademark of Critical Manufacturing.

All other trademarks are property of their respective owners.